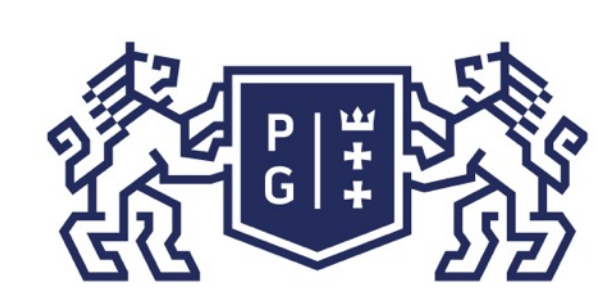




Język Java podstawy

Jacek Rumiński



Język Java podstawy

Jacek Rumiński



Katedra Inżynierii Biomedycznej,
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska





1. Wprowadzenie

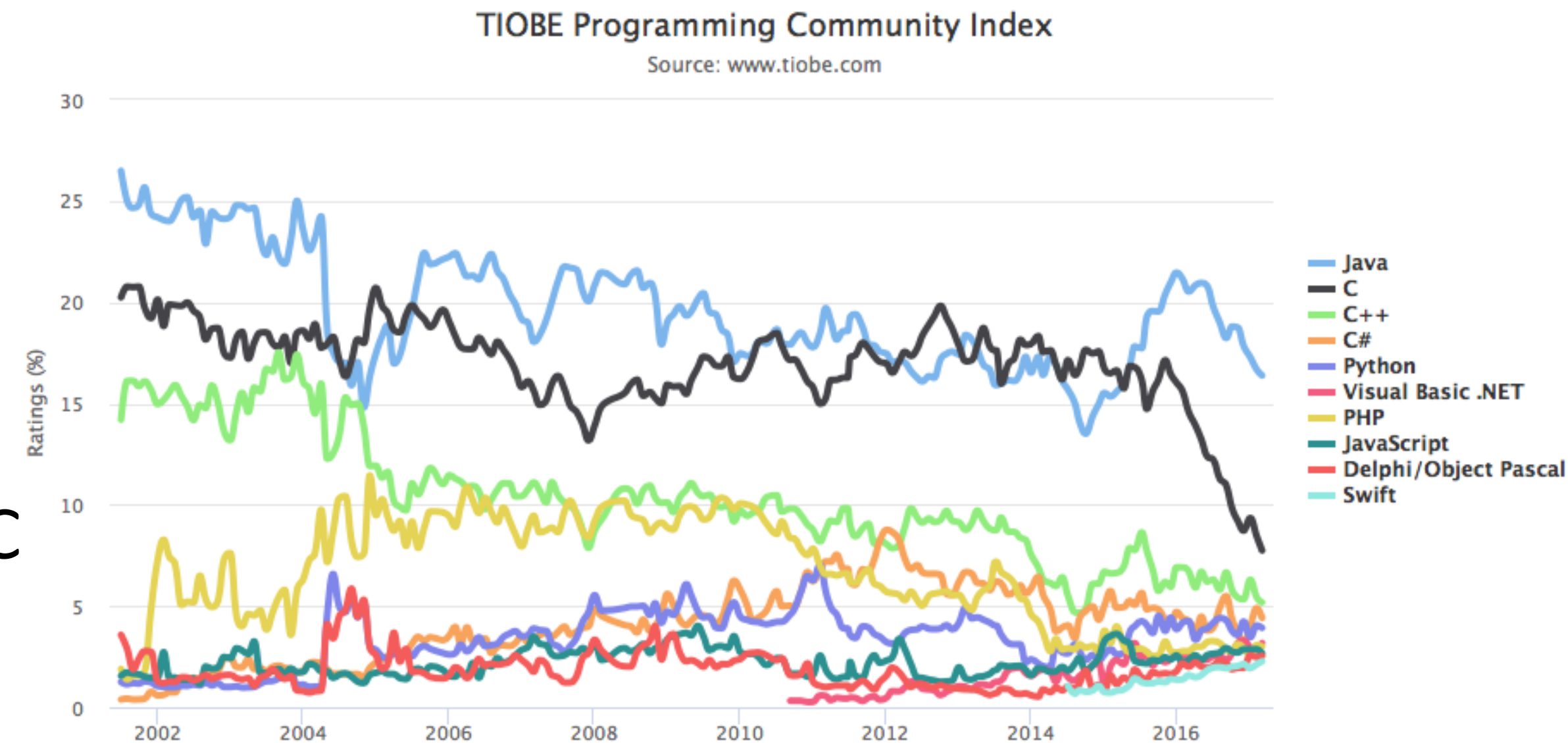
2. Konstrukcja kodu programów w Javie

3. Identyfikatory, zmienne

4. Typy danych

5. Operatory, instrukcje sterujące – instrukcja warunkowe

Język Java to jeden z najbardziej popularnych obiektowych języków programowania. Należy do grupy języków programowania bazujących na C i C++. Twórcy Javy postanowili utworzyć język wykorzystujący podstawowe elementy sterowania instrukcjami znane z języka C oraz upraszczając konstruowanie programów obiektowych względem języka C++.



Zasadniczą różnicą względem innych środowisk programowania jest to, że Java jest również platformą wykonywania kodu. Wprowadzono specjalny program zwany Maszyną Wirtualną, który tworzy dodatkową warstwę systemu operacyjnego. Dzięki temu możliwe jest wykonanie raz skompilowanego kodu na dowolnym komputerze (Mac, PC, Linux), pod warunkiem, że zainstalowano na nim Maszynę Wirtualną Javy.

Maszyny Wirtualne, kompilator, biblioteki, zintegrowane środowiska wytwarzania programów w Javie można za darmo pobrać z Internetu (<http://www.oracle.com/technetwork/java/index.html>).



1990 - Bill Joy w raporcie „Further” sugeruje SUNowi stworzenie środowiska obiektowego na bazie C++,

1991 - W ramach projektu „Green” powstaje język OAK - „Object Application Kernel” (James Gosling), przeznaczony dla aplikacji w elektronice powszechnego użytku,

1995 - zmiana nazwy na JAVA ze względu na zastrzeżenie nazwy OAK,

1996 - Pojawia się Netscape zgodny z Javą 1.0, Sun propagują darmowe środowisko JDK 1.0,

1999 - **Java 2** Nowe oblicze Javy,

do dzisiaj – rozwój bibliotek, ram projektowych, środowisk, platform,

NAZWA JAVA – kawa:



ZD 2/97 DM 9,-
ZIFF DAVIS
sfr 9,20, d\$ 72, lit 12,500, Pts 900, jr 200, dkr 41, bfr 200

B 11646 **OFFICE 97**
Tips, Kniffe, versteckte Funktionen

PCpro
Office 97

..PCpro im Internet...<http://www.zdnet.de>..PCpro im Internet...<http://www.zdnet.de>..

PG
PROFESSIONELL

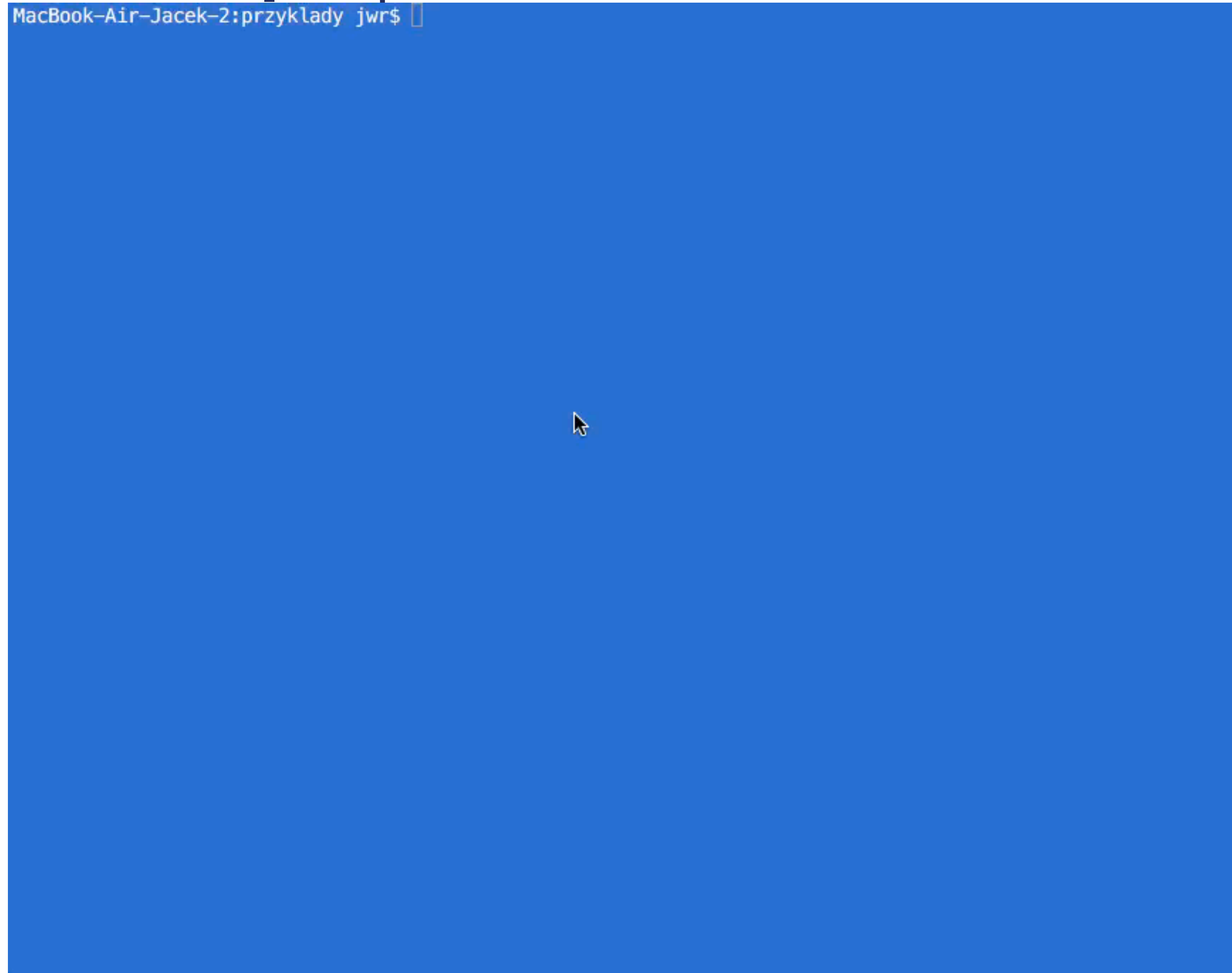
**Faszination
JAVA**
Das Ende von Windows?

- ▶ Pentium MMX
Temporekord! Die
ersten 8 PCs im Test
- ▶ Anticrash-Tools
- ▶ Noch geheim: Java-Prozessoren

Zamiast dalszych opisów zrobmy proste testy...



```
MacBook-Air-Jacek-2:przyklady jwr$
```



Kod programu: *Witamy.java*

```
/* Pierwszy program: NAZWA PLIKU TAKA SAMA jak nazwa KLASY */
```

```
public class Witamy{  
  
    public static void main(String args[]){  
        //To jest komentarz - nie jest częścią programu wykonywalnego  
        System.out.println("Pierwszy program w JAVIE");  
    }  
}  
}
```

class - początek definiowania klasy,

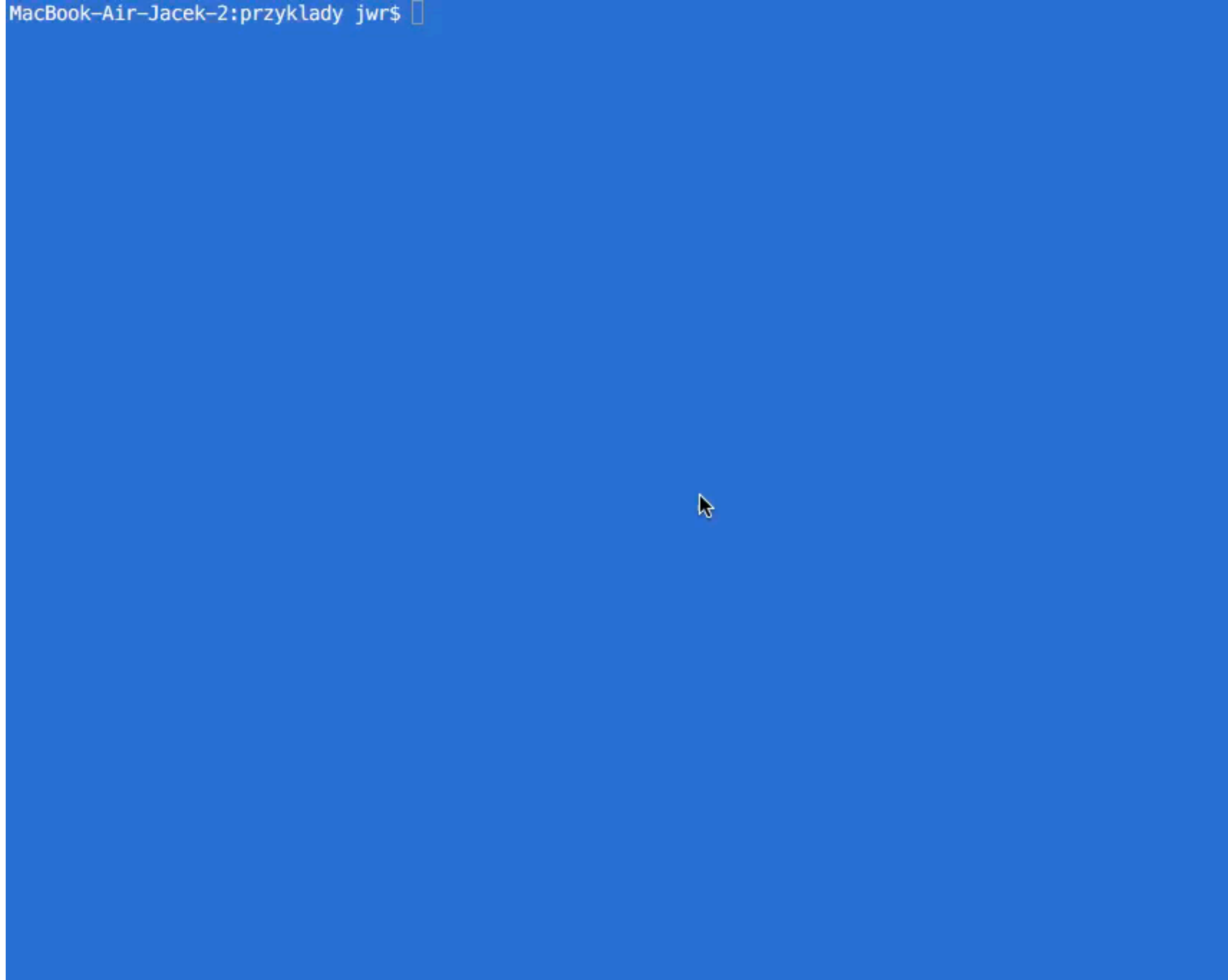
public - specyfikator oznaczający dostępność do danego elementu (*public* – dostęp z każdej innej klasy),

String – typ danych znakowych (który też jest klasą, stąd pisany z wielkiej litery).

{ } - nawiasy klamrowe do określania (grupowania) bloku kodu



```
MacBook-Air-Jacek-2:przyklady jwr$
```



Kod programu: *Witajcie.java*

```
/* Pierwszy program demonstrujący ideę użycia obiektów */  
import java.awt.*;  
public class Witajcie{  
  
    public static void main(String args[]){  
        //Utwórz obiekt istniejącej klasy Frame (okno z ramką)  
        Frame f = new Frame("Witajcie");  
        //Utwórz obiekt istniejącej klasy Button (przycisk)  
        Button b = new Button("Przycisnij mnie ...");  
        //Dodaj przycisk do okna  
        f.add(b);  
        //Ustaw rozmiar okna  
        f.setSize(500,500);  
        //Wyświetl okno  
        f.setVisible(true);  
  
    }//main  
}//class
```

Zanim jednak będziemy pisać programy z wykorzystaniem klas poznamy:

1. Jak zainstalować podstawowe narzędzia do kompilacji kodu źródłowego oraz do uruchamiania kodu po kompilacji,
2. Z czego składa się podstawowy program.

Instalacja JAVY - zanim zainstalujesz cokolwiek:

1. Co to jest JDK lub JAVA SDK:

JDK - Java Development Kit = JAVA SDK - Software Development Kit

zestaw programów, bibliotek i innych zasobów umożliwiający kompilację kodu źródłowego, wykorzystywanie standardowych zasobów (bibliotek), uruchamianie dodatkowych narzędzi lub usług, itd.

W skrócie: kompilator, maszyna wirtualna, biblioteki, narzędzia.

2. Co to jest JRE (Java Runtime Environment):

Maszyna wirtualna Javy wraz ze swoimi zasobami. Aby uruchomić program napisany i skompilowany w Javie trzeba najpierw wywołać maszynę wirtualną (program java.exe). Jeśli jesteś tylko użytkownikiem potrzebujesz wyłącznie JRE, jeśli programistą JDK = JRE +kompilator +inne.

Instalacja JAVY - zanim zainstalujesz cokolwiek:

3. Co to jest IDE (Integrated Developers Environment):

Zintegrowane środowisko programistyczne (zwykle graficzne) dostarcza edytor (z ułatwieniami, np. kolorowanie składni, numerowanie linii), debugger (analiza programu w celu poszukiwania błędów – bug), możliwość zarządzania projektem (w skład którego wchodzi wiele plików źródłowych), narzędzia projektowania interfejsu graficznego, i inne.

IDE nie jest częścią JDK. Można pisać i kompilować programy bez IDE.

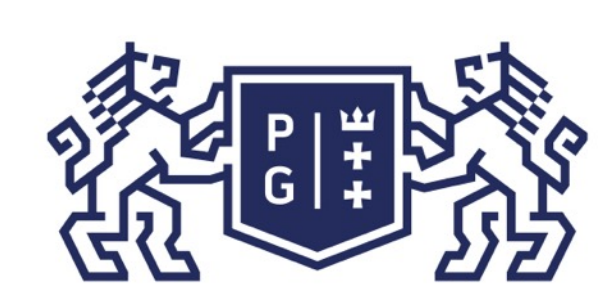
Podstawowe kroki instalowania środowiska:

1. Pobierz najnowsze JDK (wraz z JRE) z: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Pobierz najnowszą dokumentację JDK (z j.w.),
3. Opcjonalnie pobierz wybrane środowisko IDE (np. Netbeans, Eclipse, IntelliJ IDEA).
4. Zainstaluj powyższe.

UWAGA:

Niektóre środowiska IDE zawierają już JDK.

Instalacja i wykorzystanie IDE w odrębnym materiale



1. Wprowadzenie
2. Konstrukcja kodu programów w Javie
3. Identyfikatory, zmienne
4. Typy danych
5. Operatory, instrukcje sterujące – instrukcja warunkowe



```
MacBook-Air-Jacek-2:przyklady jwr$
```

```
}
```

Kod programu: *Program.java*

```
/* Z czego składa się program */
```

```
public class NazwaKlasy{
    //dane
    static int a;
    static int b;
    //funkcje
    static int sum(int c, int d){
        return (c+d);
    }
    //główna, jedna funkcja programu - main
    public static void main(String args[]){
        //Utwórz obiekt istniejącej klasy Frame (okno z ramką)
        a=2;
        b=4;
        int result=0;
        result=sum(a,b);
        System.out.println("Wynik: "+result);
    }
}
}
}
```

Komentarze

Ważnym elementem czytelnej konstrukcji kodu jest używanie dokumentacji kodu za pomocą komentarzy. W Javie stosuje się trzy podstawowe typy komentarza:

➤ **komentarz liniowy (jedna linia):**

// miejsce na komentarz do końca linii

➤ **komentarz blokowy (wiele linii):**

/*

miejsce na komentarz w dowolnym miejscu bloku

*/

➤ **komentarz dokumentacyjny (wiele linii)**

/**

miejsce na komentarz w dowolnym miejscu bloku

*/

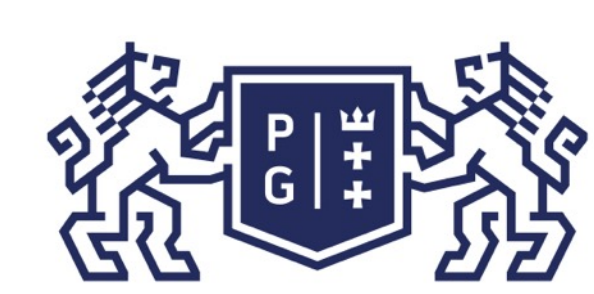
Procedura „krok po kroku”:

1. Napisanie z pomocą dowolnego edytora tekstu np. Notatnik, Notepad++, itp. (lub programu typu IDE) kodu źródłowego programu zawierającego klasę publiczną o nazwie takiej samej (dokładnie takiej samej z uwzględnieniem wielkości znaków) jak docelowa nazwa programu np. **NazwaKlasy**.

2. Nagranie kodu źródłowego jako pliku o danej nazwie z rozszerzeniem „.java”, np. **NazwaKlasy.java**

3. Kompilacja kodu źródłowego zawartego w pliku z rozszerzeniem **.java** do pliku docelowego o rozszerzeniu **.class** zawierającego kod pośredni np. **c:\> javac -g NazwaKlasy.java**

4. Uruchomienie w środowisku maszyny wirtualnej kodu pośredniego (uniwersalnego), np. **c:\> java NazwaKlasy**



1. Wprowadzenie
2. Konstrukcja kodu programów w Javie
3. Identyfikatory, zmienne
4. Typy danych
5. Operatory, instrukcje sterujące – instrukcja warunkowe

Identyfikatory – nazwy klas, metod, pól, pakietów, itd.

Identyfikator jest unikalny dla kompilatora, jednocześnie powinien wskazywać człowiekowi po co został utworzony (lub jakie jest jego znaczenie). Ważne jest zatem właściwe dobranie nazwanie identyfikatora!

Identyfikator tworzy się korzystając z liter (bez znaków narodowych), liczb, znaku podkreślenia „_” oraz znaku „\$”. Tworzona nazwa nie może się jednak zaczynać liczbą. Pierwszym znakiem może być litera bądź symbol podkreślenia „_” lub „\$”.

W Javie rozróżnialne są wielkie i małe litery w nazwach, tak więc nazwy: *Jedi* i *jedi* oznaczać będą dwa różne identyfikatory!

Tworząc własny kod (poza nauką, itp.) warto stosować wyłącznie nazwy w języku angielskim (globalna świat zasobów)!

Identyfikatory.

Tworząc identyfikator należy pamiętać, że nie może on mieć nazwy identycznej ze słowem kluczowym.

Słowa kluczowe to identyfikatory o specjalnym znaczeniu dla języka Java. Identyfikatory te są już zdefiniowane dla języka i posiadają określone znaczenie w kodzie programu.

Przykładowe słowa kluczowe: **public, class, for, if, int, char, switch**, itd.

Pełny wykaz słów kluczowych i inne informacje można znaleźć na stronie:

<http://docs.oracle.com/javase/8/>

(książka za darmo !)

Identyfikatory – wskazówki tworzenia

Kilka wskazówek tworzenia identyfikatorów:

- Identyfikatory klas rozpoczynamy z wielkiej litery, np. **Rycerz**;
- Nazwy pakietów, pól, zmiennych, metod, funkcji rozpoczynamy z małej litery, np. **org.jwr, wiek, wzrost, drukuj(), wyslij()**;
- Nazwy stałych (o czym później) wielkimi literami, np. **PI**.

Budując identyfikator złożony (z wielu słów) warto stosować jedną z następujących reguł:

- każde kolejne słowo zaczyna się od wielkiej litery (preferowany w Javie): np. **RycerzJedi, drukujStanArmii(), liczbaDzial**;
- każde kolejne słowo jest poprzedzone znakiem podkreślenia „_” (preferowane dla stałych), np. **LICZBA_OKIEN**

Zmienne

Zmienna to element dane typu, którego wartość może zostać zmieniona. Zmienna może być atrybutem obiektu.

Przykładowo atrybutem człowieka może być liczba zębów. Wartość tego atrybutu się zmienia w czasie. Co wiemy o takim atrybucie?

1. Możemy go nazwać (stosując reguły w Javie) np. `liczbaZebow`;
2. Atrybut ten może przechowywać wartości liczbowe od 0 (brak zębów) do 32 (typowo), np. `liczbaZebow=10`, `liczbaZebow=31`;
3. Atrybut ten jest zatem określonego typu (danych). W systemach komputerowych każdy typ danych musi być precyzyjnie określony i nazwany (słowa kluczowe). Dla naszego przykładu weźmy typ `int` (liczby całkowite), czyli: `int liczbaZebow`.
4. Każdy podstawowy typ danych oznacza jednocześnie rozmiar pamięci jaki jest rezerwowany do przechowywania wartości zmiennej danego typu, np. typ `int` – cztery bajty (4B).

Zmienne

5. Rozmiar przydzielanej pamięci (przechowywania wartości zmiennej) dla danego typu oznacza zakres możliwych wartości jakie może przyjmować zmienna, np. int, $4B=4*8\text{bitów}=32\text{bity}$ -> zakres = 2^{32} ; w Javie wszystkie liczby są ze znakiem (czyli od ujemnych do dodatnich, 1 bit na oznaczenie znaku, pozostaje 31 bitów) zatem

$$\text{MIN_VALUE} = -2^{31} = -2147483648$$

$$\text{MAX_VALUE} = 2^{31}-1 = 2147483647$$

(w symetrii równego podziału na liczby dodatnie i ujemne wartość „0” jest traktowana jako dodatnia i wówczas po każdej stronie znaku mamy 2147483648 pozycji).

6. Istnieje wiele typów danych i każdy ma swój rozmiar i możliwe wartości!

Zmienne

7. Zmiennym można przypisywać wartość (zmieniać) oraz odczytywać wartość:
przypisanie wartości (zastosowanie operatora „=”):

(set) `liczbaZebow=10;`

odczytanie wartości (podanie nazwy zmiennej):

(get) `System.out.println("Liczba zębów to: "+liczbaZebow);`

8. Zmienna może zostać ograniczona do stałej poprzez użycie dwóch słów kluczowych (specyfikatorów):

static – oznaczony element jest jeden, należy do danej klasy, czyli nie trzeba tworzyć obiektu, aby z takiego elementu korzystać,

final – oznaczony element ma ostateczną (kończącą) definicję, czyli przypisana wartość nie może ulec zmianie,

przykładowo (public – stała dostępna dla każdego kodu):

```
public static final int MAX_LICZBA_ZEBOW=32;
```



1. Wprowadzenie
2. Konstrukcja kodu programów w Javie
3. Identyfikatory, zmienne
4. Typy danych
5. Operatory, instrukcje sterujące – instrukcja warunkowe



Zapraszamy na 2 część w kolejnym materiale

